Introduction to Python I (Exercises 05) (Sample answers)

Remember that you can come out with a different way to solve the exercises

At the beginning while you are getting acquainted with programming and Python as a language your objective is to produce a suitable RESULT. As you get more experience, you will be able to apply your python knowledge to write elegant code. But for the time being focus on the results.

1) Write a program that defines the following dictionary:

```
student_marks = {'Joe':15, 'Mary':20, 'Ralph':18, 'Annita':19}
```

1.1 Just print t the dictionary (in dictionary form)
`{'Joe':15, 'Mary':20, 'Ralph':18, 'Annita':19}`

1.2 Add the key pair value `{'Rob':16}` to the dictionary

1.3 Assign the mark 17 to Joe

1.4 Write an If statement that checks for the existence of student 'Mary' in the dictionary. Print a message that lets you know if 'Mary' is or is not in the dictionary.

Do the same logic, but this time prompt the user for a student name. Then apply the same logic as above.

1.5 Print the elements of the dictionary, first the key and then the value:
As in:

Joe 15
Mary 20
Ralph 18
Annita 19

```
student_marks = {'Joe':15, 'Mary':20, 'Ralph':18, 'Annita':19}

#1.1
print(student_marks)

1.2#
student_marks['Rob']=16

#1.3
student_marks['Joe'] = 17

#1.4

if 'Mary' in student_marks:
    print('Mary is in the dictionnary')
else:
    print('Mary is not in the dictionnary')
```

```
#1.5

for key, value in student_marks.items():
    print(key, value)
```

2) Write a program that creates a dictionary with student names and marks. Input the names and marks from the user (2 separate lines). Stop entering names and marks, whenever the user enters a 'Q' or 'q' in the name field.

Once the dictionary is complete, print all the values of the dictionary (name and marks, side by side).

```
my_dict = {}

while(True):

    name = input('Name: ')

    if (name == 'q' or name == 'Q'):
        break

    mark = int(input('Mark: ' ))

    my_dict[name] = mark

for name, mark in my_dict.items():
    print("{:<10} {:>10}".format(name, mark))
```

3) Write a program that reads a string and returns a table of the letters that appear in the string in alphabetical order. Besides the letter, print the number of times the letter is found in the string. Case should be ignored.
A sample output of the program when the user enters the string 'This is a String with Upper and lower case Letters', would look this this:

a: 3, c: 1, d: 1, e: 5, g: 1, h: 2, i: 4, l: 2, n: 2, o: 1, p: 2, r: 4, s: 5, t: 5, u: 1, w: 2

```
sentence = 'This is a String with Upper and lower case Letters'

my_dict = {}

for i in sentence:
        i = i.lower()

        if i != ' ':
            my_dict[i] = my_dict.get(i, 0) + 1


for i in sorted(my_dict):
    print(i, my_dict[i], sep=':', end=',')
```

4) Write a program (a tiny inventory program).

The original inventory is described by the following dictionary:

fruits = {'Apples':10, 'Bananas':20, 'Oranges':15, 'Raisins':5, 'Apricots':8}

Present the user with a menu like this one:

1. Display inventory
2. Buy Fruits
3. Stock Fruits
4. Exit

When the user takes option 1: You must display all the fruits and their current stock levels on the screen.

When the user takes option 2: You must prompt the user for the fruit he/she would like to buy.

Example: Enter the fruit you would like to buy: Bananas

If the fruit is not available let the user know and re-prompt him for another fruit.

Once the fruit type is validated, prompt the user for the amount (how many).

If the users chooses a number higher that the value currently available in inventory, let the user know and re-prompt him for another amount.

If the amount is available then take it out of inventory and make the sale.

When the user selects option 3.

Prompt the user for the name of the fruit, and the amount to be stocked.

If it is a new fruit, create its inventory record.

When the user enter the option 4.

The programs exists.

---- (SAMPLE SOLUTION ---)

```
def display_menu():
    print()
    print("*--- Fruits Store ---*")
    print()
    print("1. Display Inventory")
    print("2. Buy Fruits")
    print("3. Stock Fruits")
    print("4. Exit")

    print()
    sel = input("Enter your choice: ")

    while sel not in "1234":
        sel = input("invalid choice, please enter your choice: (1,2,3,4) ")

    return sel


def display_inventory(inventory):
    print()
    print("     Fruit     ", "Qty")
    print("---------------", "---")
    for name, qty in inventory.items():
        print("{:15s}{:3d}".format(name, qty))

    print()
```

```
def buy_fruits(inventory):
    print()

    fruitname = input("What would you like to buy? (or 'q' to exit:")

    while fruitname not in inventory and fruitname != 'q':
        print("We do not carry that fruit, make another choice or 'q' to exit)")
        fruitname = input("What would you like to buy? :")

    if fruitname != 'q':
        qty = int(input("How many would you like?: "))

        while qty > inventory[fruitname]:
            print("We currently have {} {}!".format(inventory[fruitname], fruitname))
            qty = int(input("How many would you like?: "))

        inventory[fruitname] = inventory[fruitname] - qty

        print("Thank you!, Enjoy your {} {}!".format(qty, fruitname))



def stock_fruits(inventory):
    print()

    fruitname = input("What fruit would you like to add to inventory? (or 'q' to exit:")

    if fruitname == 'q':
        return

    qty = int(input("How many {} would you like to add?:".format(fruitname)))

    inventory[fruitname] = inventory.get(fruitname, 0) + qty

    print("{} {} added to inventory!".format(qty, fruitname))
    print()



# --- Start of Program ---

inventory = {"Bananas" : 20,
             "Apples":10,
             "Oranges":15,
             "Raisins":5,
             "Apricots":8}
```

```
while True:

    sel = display_menu()

    if sel == '1':
        display_inventory(inventory)
    elif sel == '2':
        buy_fruits(inventory)
    elif sel == '3':
        stock_fruits(inventory)
    else:
        print()
        print("Thank you, come back soon!")
        break

# --- End of program ---
```